

Amendments to the Claims

1. (Previously presented) A method of executing a stored procedure in a database system, the stored procedure containing at least an expression and a database query language statement, the method comprising:

identifying the expression in the stored procedure, the expression being according to one of plural predetermined types of expressions;

generating low-level code representing the expression; and

generating object code representing the stored procedure, the object code containing the low-level code and one or more instructions representing the database query language statement.

2. (Original) The method of claim 1, further comprising directly executing the low-level code at run time to evaluate the expression.

3. (Original) The method of claim 2, wherein directly executing the low-level code is performed in place of submitting a database query language statement to evaluate the expression.

4. (Original) The method of claim 3, wherein directly executing the low-level code consumes less database system resources than submitting a database query language statement to evaluate the expression.

5. (Original) The method of claim 1, further comprising:

submitting the low-level code to an evaluator module to evaluate the expression;

and

submitting a command corresponding to the database query language statement to an access module in the database system to access data specified by the database query language statement.

6. (Original) The method of claim 5, wherein the database system has a first node containing a parsing engine and the evaluator module and a second node containing the access module, wherein submitting the command is performed by the parsing engine.

7. (Previously presented) The method of claim 1, further comprising:
storing information pertaining to a variable and a constant used in the expression with the low-level code in the object code;
executing the low-level code during execution of the stored procedure using an evaluator module; and
using the information pertaining to the variable and constant during execution of the low-level code to evaluate the expression.

8. (Original) The method of claim 1, wherein identifying the expression comprises identifying one of a conditional expression, an assignment expression, and a dynamic database query language statement.

9. (Previously presented) The method of claim 8, further comprising:
identifying a second expression in the stored procedure that is one of a conditional expression, an assignment expression, and a dynamic database query language statement;
and
generating second low-level code to represent the second expression,
wherein generating the object code comprises providing the second low-level code in the object code.

10. (Previously presented) The method of claim 1, wherein generating the object code comprises generating the object code containing the low-level code that is different from the instructions representing the database query language statement.

11. (Original) The method of claim 10, wherein generating the low-level code comprises generating assembly code.

12. (Previously presented) The method of claim 1, further comprising:
storing the object code in a predetermined location; and
accessing the predetermined location to retrieve the object code in response to
invocation of the stored procedure.

13. (Previously presented) The method of claim 12, wherein storing the object
code in the predetermined location comprises storing the object code in a stored
procedure table.

14. (Previously presented) The method of claim 12, further comprising:
executing the object code, wherein executing the object code comprises submitting the
low-level code to an evaluator module to execute the low-level code; and
executing the one or more instructions representing the database query language
statement without submitting the one or more instructions to the evaluator module.

15. (Original) An article comprising at least one storage medium containing
software that when executed cause a database system to:
generate object code corresponding to a stored procedure having at least a first
type expression and a second type expression, the first type expression selected from the
group consisting of a conditional expression, an assignment expression, and a dynamic
database query language statement;
create a predetermined type of code corresponding to the first type expression;
and
provide the predetermined type of code in the object code to represent the first
type expression; and
provide one or more instructions representing the second type expression in the
object code, the instructions different from the predetermined type of code.

16. (Original) The article of claim 15, wherein the software when executed causes
the database system to execute the predetermined type of code directly at run time to
evaluate the first type expression.

17. (Previously presented) The article of claim 16, wherein the software when executed causes the database system to evaluate a database query language statement in the second type expression in evaluating the second type expression.

18. (Original) The article of claim 17, wherein the software when executed causes the database system to:

provide the first type expression to an evaluator module to generate the predetermined type of code; and

provide the second type expression to an object code generator to generate the instructions representing the second type expression.

19. (Original) The article of claim 18, wherein the software when executed causes the database system to provide the predetermined type of code to the object code generator to add to the object code.

20. (Original) The article of claim 15, wherein the software when executed causes the database system to:

provide the predetermined type of code to an evaluator module to evaluate the expression; and

submit a command corresponding to a database query language statement in the second type expression to an access module of the database system to access data specified by the database query language statement.

21. (Original) The article of claim 15, wherein the predetermined type of code corresponding to the first type expression includes machine-level code, and wherein the instructions representing the second type expression includes C code.

22. (Original) The article of claim 15, wherein the second type of expression comprises a Structured Query Language (SQL) statement.

23. (Original) A database system comprising:
a plurality of nodes;
an evaluator module in a first one of the plurality of nodes;
an access module in a second one of the plurality of nodes, the access module to manage access to a portion of data stored in the database system; and
a controller in the first node adapted to execute a stored procedure object code, the object code containing a first type of code to represent an expression that is one of a conditional expression, an assignment expression, and a dynamic statement, the object code containing a second, different type of code to represent a database query language statement,
the controller adapted to submit the first type of code to the evaluator module to evaluate the expression, and
the controller adapted to submit a command corresponding to the database query language statement to the access module.

24. (Original) The database system of claim 23, wherein the controller comprises a parsing engine.

25. (Original) The database system of claim 23, wherein the first type code comprises machine-level code.

26. (Original) The database system of claim 23, wherein the first type code contains information identifying a type of the expression and a variable and constant used by the expression.

27. (Original) An article comprising at least one storage medium containing instructions for use in a database system, the instructions when executed causing the database system to:

access object code in response to invocation of a stored procedure, the object code containing first type code representing an expression and second type code representing a database query language statement;

submit the first type code to an evaluator module to evaluate the expression; and
submit a command corresponding to the database query language statement to an
access module to access data specified by the database query language statement.

28. (Original) The article of claim 27, wherein the instructions when executed
cause the database system to:

generate the first type code for the expression, the expression being one of a
conditional expression, assignment expression, and dynamic statement; and
provide the first type code in the object code.

29. (Original) The article of claim 28, wherein the instructions when executed
cause the database system to:

generate the second type code which is different from the first type code; and
provide the second type code in the object code.